REMARKS

Favorable reconsideration and allowance of the claims of the present application, as amended, is respectfully requested.

In the present Office Action, the Examiner rejected Claims 31 and 34 under 35 U.S.C. § 103(a) as allegedly unpatentable over Bacon et al. U.S. Patent No. 6,430,538 (hereinafter " Bacon") in view of Acosta et al. U.S. Patent No. 6,166,729 (hereinafter "Acosta") and, in further view of Shi (U.S. Patent No. 5,381,534).

Applicants respectfully disagree.

Independent Claim 34 sets forth subject matter that is patentably distinct from the cited combination in several important respects.

The present invention as set forth in independent Claim 34, is directed to a method of distributing work through a cluster of workstations for efficient distributed processing, the cluster having a plurality of workstations interconnected over a network. In one aspect, the method includes steps of:

> receiving a work request at a first processing node;
> classifying, at said first processing node, the work request into one or more tasks;
> and, assigning said one or more tasks to one or more router queues associated with respective router devices at said first processing node, a router device for receiving and distributing a specific task of a particular class of work, each said router queue associated with a work task at a different phase of completion;

Applicants note an amendment to Claim 34, to clarify that the router is a device for receiving and distributing a specific task of a particular class of work. This is clearly set forth on

page 10, line 20-32 and all of page 11 of applicants originally filed specification which speaks to the types of router queues and the work task they handle with no new matter being entered. Further, Claim 34 is being further amended to clarify that the each router queue is associated with a work task at a different phase of completion which enables work at a different phase of completion to flow through said cluster of workstations. Respectfully, no new matter is being entered as full support is found in the application as originally filed e.g., page 11, lines 5-27. While the Examiner has indicated that Bacon at col. 4, lines 38-47 allegedly teaches the classification of a work request into tasks and, at a processing node, and assignment of work tasks to router devices, with a router capable of handling a specific task of a particular class of work, Applicants respectfully question this application of Bacon and respectfully traverse:

1. From the Examiner's interpretation, the server device (Bacon element 110, Fig. 2) having engines 115 are suggestive of the first processing node that receives the initial work request of the present invention that includes classifier 104 and plural routers 106a, et seq., to which work tasks are assigned. Bacon does not teach nor suggest the provision of a router device for receiving and distributing a specific task of a particular class of work. Moreover, Bacon does not teach or suggest as claimed in amended Claim 34, a router queue associated with a work task at a different phase of completion which enables work at a different phase of completion to flow through said cluster of workstations as in the present invention. Bacon, rather teaches network connection of a client with a server. While a client/server communication may necessarily require or encounter a router device on the network, there is no teaching or suggestion in Bacon as to specialized router devices as in the present invention for receiving and distributing a

specific task of a particular class of work. Moreover, Bacon does not teach or suggest as claimed in amended Claim 34, a router queue associated with a work task at a different phase of completion to enable work at a different phase of completion to flow through the cluster of workstations.

At best, Bacon teaches an activity determining step from a process definition file (element 107 stored in its database 125, see Fig. 1 of Bacon) for determining when an activity may be started in which case Bacon's engine 115 routes a given work item 117 to the appropriate actors, such as agents 120, clients 130...where an activity is performed (See Bacon at col. 4, lines 51-55). Moreover, as stated in Bacon, at col. 5, lines 1-5, Bacon's work items 117 are "stored in the database" 125 after each activity is performed, and "read from the database" 125 each time a work item 117 is sent to an actor (a "work item" in Bacon is a representation of a document or information being passed through a business process, e.g., in a collaborative networked environment- see Bacon at col. 2, lines 22-24 and server's database 125 is an "object-oriented" database see Bacon at col. 5, lines 38-48). Thus, there is no router device or queue dedicated to receiving and distributing a specific task of a particular class of work as now claimed in amended Claim 34.

Further, the method as claimed in Claim 34, includes steps of:

> dispatching said assigned one or more tasks for execution at a workstation at a second processing node having an execution module residing therein, the execution module at said second processing node comprising one or more initiators for instantiating one or more objects to execute a respective work task, said initiators dynamically registering with a router to indicate readiness

to accept work for processing...

2. It appears from the Examiner's interpretation, that the client or agent device (Bacon element 120, 130, Fig. 1) are suggestive of the workstation at a claimed second processing node having an execution module residing therein comprising one or more initiators for instantiating one or more objects to execute a respective work task. However, applicant is hard pressed to find the teaching of an initiator at the client or agent (of Bacon) that instantiates one or more objects to execute a respective work task. Rather, it seems the engines 115 of the Bacon server 110 (Bacon, Fig. 1-2) provide the clients with information enabling user access to the object at the database 125 (see Bacon at last paragraph bridging col. 5 and col. 6). That is, Bacon at col. 6, lines 48-50 teaches the server 110 sending a work item event message to a client of interest 130 indicating that a work item object 117 has been scheduled for that activity and further states that "the work item-related information is determined from the engine 115, the definition 107, the state of the workflow, and the database 125".

That is, respectfully, applicants interpret Bacon's server 110 (first processing node) as being the "initiator" because:

In Bacon, notwithstanding teaching of the commonly implemented CORBA (object request broker) functionality, the server 110 and engine 115 process a definition file 107 which provides all of the information of which node (client) is to execute the task. That is, the server 110 functions as an "initiator" for providing one or more objects to execute a respective work task as the server 110 causes a selected work item object to be distributed to the client which entails (using a Java implementation) reading the object 117 from database 125 using its object

identification and establishing IOR object references on the client 130 to reference an ORB which in turn references the object implementation of the work item object 117 on the server 110. Thus, the task (work) of a client is "initiated" by the server for processing at the server which communicates with the client (through a browser interface) the scheduled activity at that client.

While the Examiner interprets Bacon broadly, Applicants are still hard pressed to find a teaching in Bacon of initiators dynamically registering with a router to indicate readiness to accept work for processing as claimed in Claim 34. In this regard, the Examiner cites Bacon at col. 6, lines 65-66 as teaching such dynamic registering to indicate readiness to accept work. This particular passage however, appears misplaced as this passage in Bacon speaks to the client (user) "selecting a work item of interest from its in-box" indicating it is ready to begin the activity. This pre-supposes and assumes that the client has first received (in its "in-box") a notification from the server 110 that a work item (task) for processing by that client is available. That is, as mentioned earlier, server 110 sends a work item event message to a client of interest 130 indicating that a work item object 117 has been scheduled for that activity and, in response to receiving a new work item event, the client requests from the server 110 a refresh of the client's in-box which prompts the server to obtain the "in-box" information from the database 125 and send it to the client. The in-box information includes the names of the work items assigned to the client and may include other work item related information such as corresponding priority information, URL references to HTML pages, or the like.

Applicant's respectfully submit that this Bacon teaching does not rise to the level of the recitation as claimed in Claim 34, namely, initiators dynamically registering with a router to

indicate readiness to accept work for processing. The client's in Bacon do not actively "register" with a server to indicate "readiness", but rather, listen on a socket for receipt of the server message in its in-box. Thus, the Bacon client in this context is functioning passively, as a listener, responding to the server's initiation (a server message received at the client's in-box). Only by the client responding to the server by requesting from the server 110 a refresh of the client's in-box which prompts the server to respond with further information, is this an indication of a client's "readiness".

This is further bolstered by the recitation in Bacon at col. 6, lines 53-64, which is directed to the Bacon flow engine 115 (Bacon Figs. 1, 5) interpreting the definition 107, along with other information stated above, and subsequently sending a set of possible next scheduled activities to the client for selection via the "in-box".

Respectfully, there is no teaching or suggestion in Bacon of initiators dynamically registering with a router to indicate readiness to accept work for processing.

Further, the method as claimed in Claim 34, includes:

> ...said objects instantiated by an initiator with a generic class name passed to the initiator by said router but having a different implementation specific to a node in which said initiator resides to enable use of system specific resources and enable a single version of an application to run on each node; and,

3. It appears from the Examiner's rejection citing Col. 7, lines 12-25, 45-53 and col,. 8, lines 1-12 that the mere use of an ORB or CORBA functionality teaches instantiation using a generic class name but having a different implementation specific to a

node in which said initiator resides. However, as argued herein, Bacon does not teach presence of object thread instantiators at a second processing node; nor does Bacon teach providing the initiator with a generic class name passed to the initiator by the router, as set forth in amended Claim 34, but having a different implementation specific to a node in which said initiator resides. Moreover, the cited passages seem to highlight Bacon's use of a referenced predefined Java applet, script and HTML page that is loaded at the client that corresponds to the client application for the activity being performed. This, respectfully does not rise to the level of an object instantiator at a workstation of a second processing node for instantiating an object using a generic class name but having a different implementation specific to that node in which said initiator resides to enable use of system specific resources and enable a single version of an application to run on each node. In fact, Bacon teaches away as it is the execution of an HTML page and pre-defined applet that provides specificity for interacting with and processing an object.

Further, the method as claimed in Claim 34, includes:

> ... upon completion of said respective work tasks, each said one or more initiators providing to said respective router the completed work task at said first processing node and providing system specific statistics data associated with said initiator; and,
> computing performance statistics of a router queue and said one or more initiators, a performance statistic including a total response time from dispatch of a work task from that router queue at said first processing node to an initiator at said second processing node, and the receipt of the completed work task at the router queue from that initiator at said second processing node, said

total response time used to determine the performance of an initiator and categorize the initiator performance for determining said one or more initiators best suited to execute said one or more tasks; and,
queuing ready initiators at a respective router device based on said categorized initiator performance, wherein said best performing ready initiators are given priority for receiving new tasks from a respective router

4. In the rejection of Claim 34, the Examiner cites Acosta at col. 14, lines 33-50 as providing a teaching of determining performance statistics associated with one or more router queues and adding more workstations based on performance statistics of the one or more router queues. Applicants disagree. Acosta's teaching is to maintain a certain quality of level and adding more system resources to maintain that level based on queue activity at a single server. That is, as taught in Acosta, the actual "performance statistic" being tracked comprises the queue size (i.e., a number of images waiting in the queues), an average queuing latency, the average time to live status of images in the queues, and dwell time ratios among the queues. This teaching has nothing to do with determining the performance of an initiator and categorizing the initiator performance at said second node. Acosta performs this "queue monitoring" only for purposes of keeping the statistics "within desired values" at that specific processing node (server), by adjusting variable parameters of system operation (e.g., adds more servers, adds more bandwidth, adds more memory, and so forth).

In distinction, Claim34 is being amended to clarify that the performance statistic of the invention includes a total response time from dispatch of a work task from that router queue at said first processing node to an initiator at said second processing

node, and the receipt of the completed work task at the router queue from that initiator at said second processing node. The Examiner acknowledges that the combination of Bacon and Acosta does not teach the claimed total response time measurement (from dispatch of a work task from that router queue at said first processing node to an initiator at said second processing node, and the receipt of the completed work task at the router queue from that initiator at said second processing node) and cites Shi at col. 25, lines 3-9 as making up this deficiency.

Applicants respectfully disagree.

Shi generically teaches the evaluation of the quickest performing computer in order to more effectively achieve "load balancing" (as articulated in Shi at col. 25, lines 3-9). However, this motivation is also the motivation for monitoring the flow engine as taught in Acosta. While beneficial, the present application does not measure or speak to "load balancing" at all but rather to ascertain a good performing initiator running at the second processing node. The Examiner seems to rely on a basis of "load balancing" as the motivation to combine these references. But, rather, the motivation of the present invention is to provide distributed processing without the need to have special configuration information related to a particular workstation and, to determine the performance of initiators and categorize the initiator performance for purposes of determining one or more initiators best suited to execute subsequent one or more tasks.

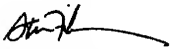
In view of the foregoing, Applicants respectfully submit that amended independent Claim 34 obviates all the rejections based on 35 U.S.C. §103(a).

It is thus respectfully requested that the Examiner's rejections of these claims and

remaining Claim 31 dependent thereon, be withdrawn.

In view of the foregoing remarks herein, it is respectfully submitted that this application is in condition for allowance. Accordingly, it is respectfully requested that this application be allowed and a Notice of Allowance be issued. If the Examiner believes that a telephone conference with the Applicants' attorneys would be advantageous to the disposition of this case, the Examiner is requested to telephone the undersigned, Applicants' attorney, at the following telephone number: (516) 742-4343.

Respectfully submitted,



Steven Fischman
Registration No. 34,594

SCULLY, SCOTT, MURPHY & PRESSER, P.C.
400 Garden City Plaza, Suite 300
Garden City, New York 11530
(516) 742-4343

SF:gc